# Docker: how to "package" your development environment

dbCafé - 8 luglio 2015

Cristian Consonni

# Quick survey

- How many people have heard about Docker before this dbCafé?
- How many people have tried Docker?
- How many people are using it ~~every day~~ very often?

# Part 1

## Introduction to Docker

(derived from: http://bit.ly/docker-jugtaa-1 and http://bit.ly/docker-jugtaa-2 by Mario Alexandro Santini and Cristian Consonni)

# LXC (LinuX Containers)

«**LXC** (**Linux Containers**) is an operating-system-level virtualization environment for running multiple isolated Linux systems (containers) on a single Linux control host.»

(source: https://en.wikipedia.org/wiki/LXC)

- From inside, it looks like a VM
- From outside it looks like normal processes
- "*chroot on steroids*"

(source Jérôme Petazzoni, DevOps at dotCloud, http://bit.ly/1flGjit)

# cgroups and namespaces

«**cgroups** (abbreviated from **control groups**) is a Linux kernel feature that limits, accounts for and isolates the resource usage of a collection of processes.»

(source: https://en.wikipedia.org/wiki/Cgroups)

Resources:

- CPU
- memory
- disk I/O
- network
- etc.

# aufs

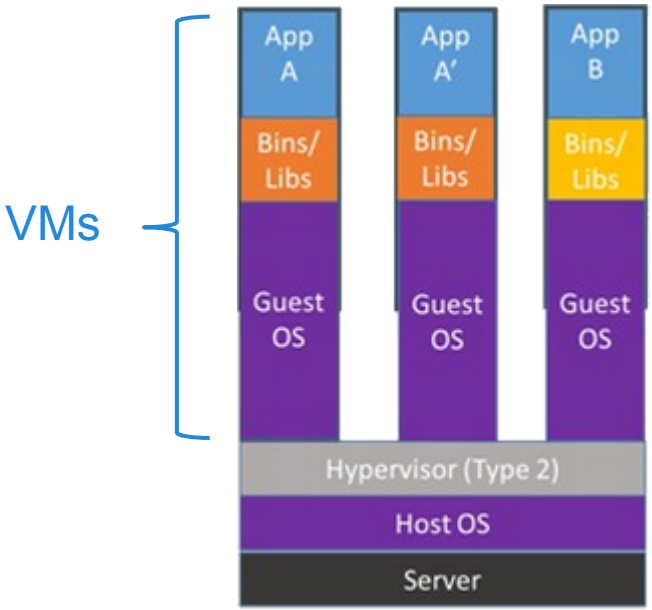
«**aufs** (short for **advanced multi layered unification filesystem**) implements a union mount for Linux file systems.»

(source: https://en.wikipedia.org/wiki/Aufs)

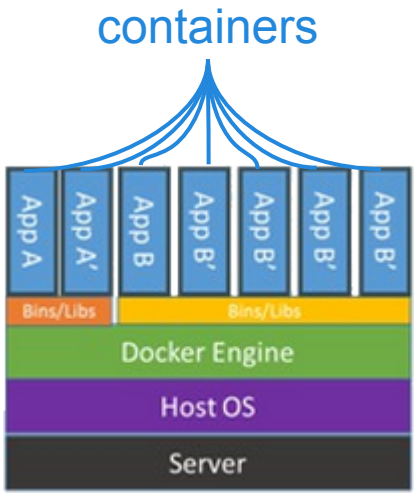
Union mount allow several directory to be mounted at the same mount point, appearing to be a single file system. Aufs supports copy-on-write.

# Virtualization vs Containers

## Virtualization




## Container




(source: http://anandmanisankar.com/posts/container-docker-PaaS-microservices/)

# Why Containers?

- **Speed:** boots in seconds

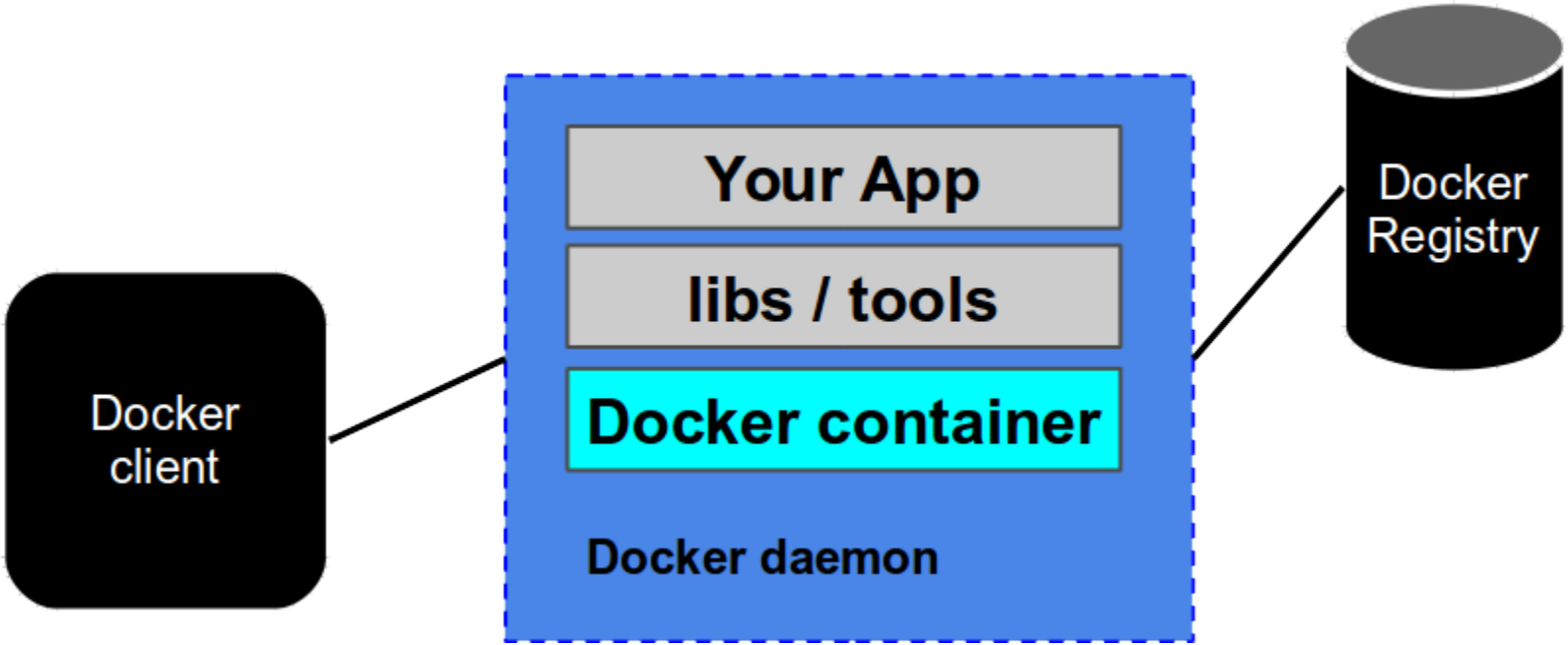
- **Resource/memory footprint**: 100-1000 containers can run on a single machine

# What is Docker? (I)

«**Docker** is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating-system-level virtualization on Linux»

(source: https://en.wikipedia.org/wiki/Docker_(software))

# What is Docker? (II)



# What is Docker? (III)

Huge success:

- project started by dotCloud in January 2013
- Open sourced in March 2013
- Written in Go
- (as of July 2015) on GitHub 990+ contributors, 14000+ PR, 16500+ commits

# Dockerfile (I)

```
1 FROM java:7-jre
2
3 ENV CATALINA_HOME /usr/local/tomcat
4 ENV PATH $CATALINA_HOME/bin:$PATH
5 RUN mkdir -p "$CATALINA_HOME"
6 WORKDIR $CATALINA_HOME
7
```

# Dockerfile (II)

```
 8 # see https://www.apache.org/dist/tomcat/tomcat-8/KEYS
 9 RUN gpg --keyserver pool.sks-keyservers.net --recv-keys \
10 05AB33110949707C93A279E3D3EFE6B686867BA6 \
11 07E48665A34DCAFAE522E5E6266191C37C037D42 \
12 ...
13
14 ENV TOMCAT_MAJOR 6
15 ENV TOMCAT_VERSION 6.0.44
16 ENV TOMCAT_TGZ_URL https://www.apache.org/dist/tomcat/
17 tomcat-$TOMCAT_MAJOR/v$TOMCAT_VERSION/bin/
18 apache-tomcat-$TOMCAT_VERSION.tar.gz
```

# Dockerfile (III)

```
19
20 RUN set -x \
21     && curl -fSL "$TOMCAT_TGZ_URL" -o tomcat.tar.gz \
22     && curl -fSL "$TOMCAT_TGZ_URL.asc" -o tomcat.tar.gz.asc
23     && gpg --verify tomcat.tar.gz.asc \
24     && tar -xvf tomcat.tar.gz --strip-components=1 \
25     && rm bin/*.bat \
26     && rm tomcat.tar.gz*
27
28 EXPOSE 8080
29 CMD ["catalina.sh", "run"]
```

# DockerHub

- Docker Hub: public and private registry
- Docker registry: local registry (can be also remote or a company registry)
- Docker Trusted Registry: service provided by Docker, Inc.

# Part 2

Docker hands-on

# Install Docker

- Ubuntu:

```
$ wget -qO- https://get.docker.com/ | sh
```

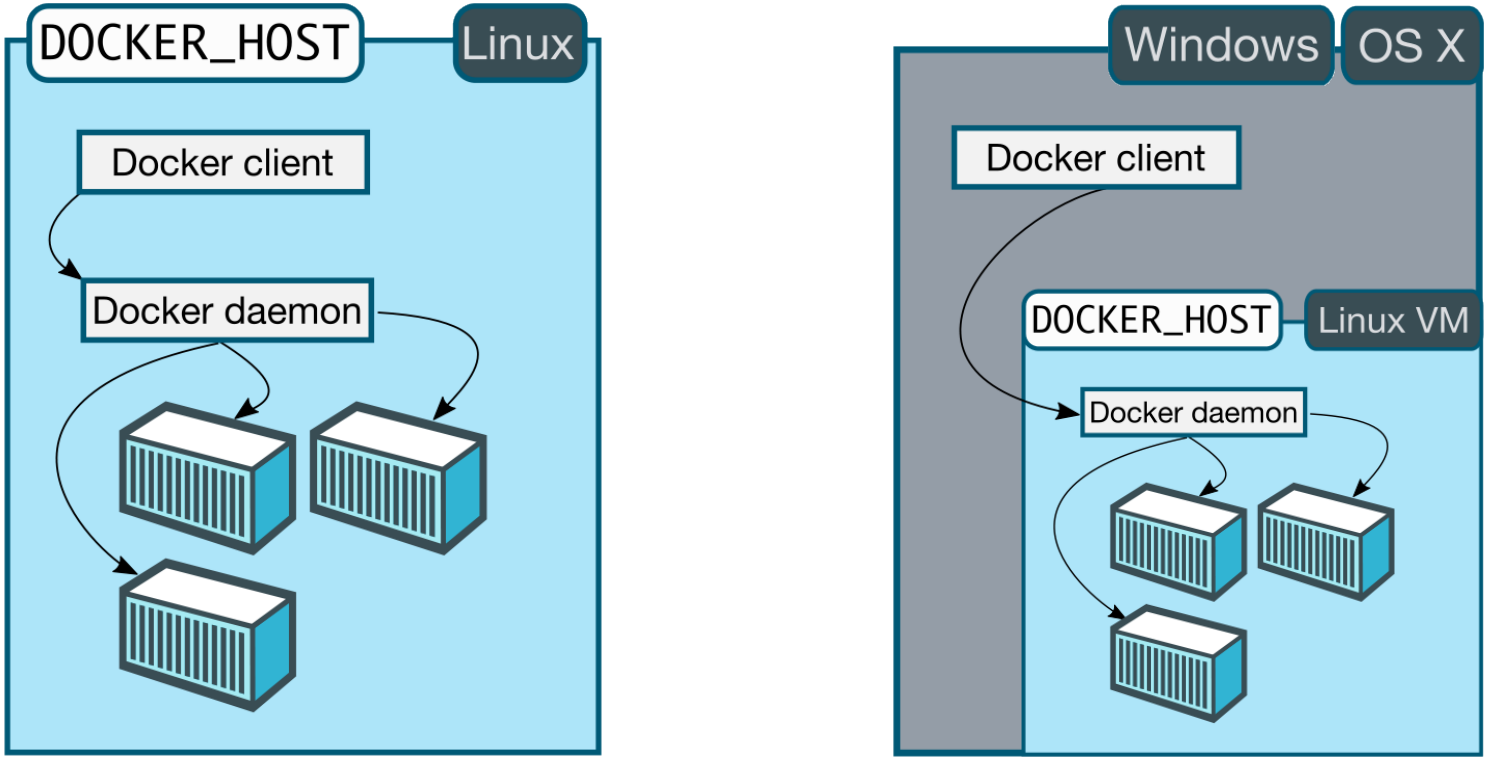
(version >= 14.04  for previous version see the prerequisites at: https://docs.docker.com/installation/ubuntulinux/)

(`$ sudo usermod -aG docker ubuntu`)

(also packaged: `$ sudo apt-get install docker.io`)

- Mac OS X: Boot2Docker (CLI) o Kitematic (GUI)
- Windows: Boot2Docker (CLI)

# Docker on Linux and Mac/Win



# Docker client commands (I)

`docker <subcommand>`

- `pull NAME[:TAG]`

downloads the `NAME[:TAG]` from DockerHub

- `run IMAGE [COMMAND]`

launches a container with the `IMAGE` and execute `COMMAND`

- `build PATH | URL | -`

builds a new image from the Dockerfile in `PATH`

# Docker client commands (II)

- `ps [-a]`

list running containers, with `-a` list all containers

- `images`

list images that are available locally

- `rm`

remove a container

- `rmi`

remove an image (you can not remove images you are using)

# Let's try: `hello world` (II)

```
docker ps -a
docker images
docker rm …
docker rmi …
```

(https://asciinema.org/a/b7j71yim7dewcka2w421s0398)

# Something more: launch `bash` on Ubuntu

```
docker run -t -i ubuntu:latest
/bin/bash
```

(https://asciinema.org/a/5v36v9rxpvp4ywhqb4gk0bis5)

```
docker ps
```

(https://asciinema.org/a/333daetaiq7doh6cglbh8t3c8)

# `docker run` options (I)

- `-p <container_port>:<host_port>`

Links the port `<container_port>` to `<host_port>`

- `-t`

Allocates a pseudo-tty

- `-i`

keep `STDIN` open even if it is not attach to the terminal in use

# `docker run` options (II)

- `--rm`

Automatically removes the container when it is terminated

- `-v <local_dir>:<container_dir>`

Mounts a local dir in the specified container dir (e. g. `$PWD:/mnt`)

# Conclusions

- Docker uses *Linux Container* (`LXC`) to automate the deploy process
- Containers provide development environment that can be standardized  completely (no *“works on my machine”* anymore)
- We want to deploy with the same ease both on the developer/researcher laptop and in production
- Docker is heavily developed and things are moving fast

# Coming soon...

- How to write a Dockerfile: `Dockerfile commands`
- How to build a Docker image
- How to package a simple web application
- How to build a multi-container app
- Orchestration and much more...

# Thank you!